Privacy preserving and cost optimal mobile crowdsensing using smart contracts on blockchain

Chatzopoulos, D.

IEEE
2018

Chatzopoulos , D , Gujar , S , Faltings , B & Hui , P 2018 , Privacy preserving and cost optimal mobile crowdsensing using smart contracts on blockchain . in 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS) . IEEE International Conference on Mobile Ad-hoc and Sensor System , IEEE , pp. 442-450 , IEEE International Conference on Mobile Ad Hoc and Sensor Systems , Chengdu , China , 09/10/2018 . https://doi.org/10.1109/MASS.2018.00068

http://hdl.handle.net/10138/307211
https://doi.org/10.1109/MASS.2018.00068

other
acceptedVersion

Privacy Preserving and Cost Optimal Mobile Crowdsensing using Smart Contracts on Blockchain

Dimitris Chatzopoulos*, Sujit Gujar&, Boi Faltings$, and Pan Hui*#
dcab@cse.ust.hk, sujit.gujar@iiit.ac.in, boi.faltings@epfl.ch, panhui@cse.ust.hk
*HKUST, &IIIT Hyderabad, $EPFL, #University of Helsinki

***Abstract*—The popularity and applicability of mobile crowdsensing applications are continuously increasing due to the widespread of mobile devices and their sensing and processing capabilities. However, we need to offer appropriate incentives to the mobile users who contribute their resources and preserve their privacy. Blockchain technologies enable semi-anonymous multi-party interactions and can be utilized in crowdsensing applications to maintain the privacy of the mobile users while ensuring first-rate crowdsensed data. In this work, we propose to use blockchain technologies and smart contracts to orchestrate the interactions between mobile crowdsensing providers and mobile users for the case of spatial crowdsensing, where mobile users need to be at specific locations to perform the tasks. Smart contracts, by operating as processes that are executed on the blockchain, are used to preserve users' privacy and make payments. Furthermore, for the assignment of the crowdsensing tasks to the mobile users, we design a truthful, cost-optimal auction that minimizes the payments from the crowdsensing providers to the mobile users. Extensive experimental results show that the proposed privacy preserving auction outperforms state-of-the-art proposals regarding cost by ten times for high numbers of mobile users and tasks.**

I. Introduction

The wide dissemination of smartphones that are programmable and employed with sensors gave birth to *crowdsensing* applications such as environment monitoring, mobile social recommendations, public safety and others. Mobile crowdsensing is a paradigm that utilizes the ubiquitousness of the mobile users who are carrying smartphones and can collect and process data. *Crowsensing Service Providers* (CSPs) request sensing *tasks* to *mobile users* (MUs) who deliver these tasks in order to get paid. Crowdsensing tasks can be categorized based on characteristics inherent to the tasks or the participants[1]. Two usual dimensions are event based vs. continuous, and spatial vs. non-spatial. These dimensions are independent of each other, and any combination is possible.

In this work, we focus on *event-based spatial crowdsensing tasks* that are associated with geographic locations where the mobile users perform them [1], [2]. The challenges are twofold: *(i)* the mobile users are sensitive about the secrecy of their locations and may not participate to avoid any leakage. Also, they may even try to spoof their locations to avoid the cost of moving the required locations. *(ii)* A second challenge is the calculation of the payments to MUs for their participation. The *participation cost* of each user is private information and depends on several factors. As a consequence, mobile users are motivated to misreport their actual costs to obtain higher payment, and hence incentives are needed. Truthful auctions are designed in such a way to force participants to report their true participation cost. This feature enables optimal task assignment to the participants in such a way to minimize the payments to the employed mobile users [3].

We consider participants who are not willing to reveal their identities and locations regardless of the number of the tasks they have delivered. Although Internet service providers (*ISPs*) are aware of users' identities and locations, they are not allowed to reveal them to third-parties [4]. We propose to use the capabilities of ISPs supplemented by smart contracts over *blockchains* to design a system for privacy-preserving crowdsensing that minimizes CSPs' cost. We propose a model where CSPs send crowdsensing requests to an ISP who transforms them into tasks and runs a cost-optimal auction to the suitable cells to allow the MUs on these cells to express their interest in the tasks via truthful bidding. The ISP is assisted by a blockchain, similar to Ethereum [5] and Hawk [6] or Hyperledger Fabric [7]. To build such crowdsensing system, we address the following questions:

- $\mathcal{Q}_1$: How to ensure a CSP that the data has been submitted by users at the indicated locations?
- $\mathcal{Q}_2$: How to preserve the privacy of mobile users from CSPs, even if they have submitted location-specific data?
- $\mathcal{Q}_3$: How to assign crowdsensing tasks to mobile users who are interested in subsets of tasks in a cost-optimal way and incentivize them to report their costs truthfully?

For $\mathcal{Q}_1$ and $\mathcal{Q}_2$, we leverage the confidentiality assurance from ISPs. ISPs guarantee the execution of CSPs' tasks at the desired locations. To build such trust across CSPs, ISPs, and MUs, we use a blockchain and *smart contracts*. To address $\mathcal{Q}_3$ we design an auction using game theory.

Why blockchain? *Blockchain* is a distributed mechanism that stores data in the form of transactions and can offer additional functionalities such as *transactional privacy* and *smart contracts*. It is maintained by interconnected nodes that are responsible for securing the network, and keeping everyone in the system in sync. Anyone interested in maintaining a blockchain, and, as a consequence, in having access to the stored data can partake. Blockchains have been used in mobile environments such as for automated payments between mobile devices in cooperative application execution scenarios [8]

[1]We are using the terms "mobile users" and "participants" interchangeably and depending on the context.

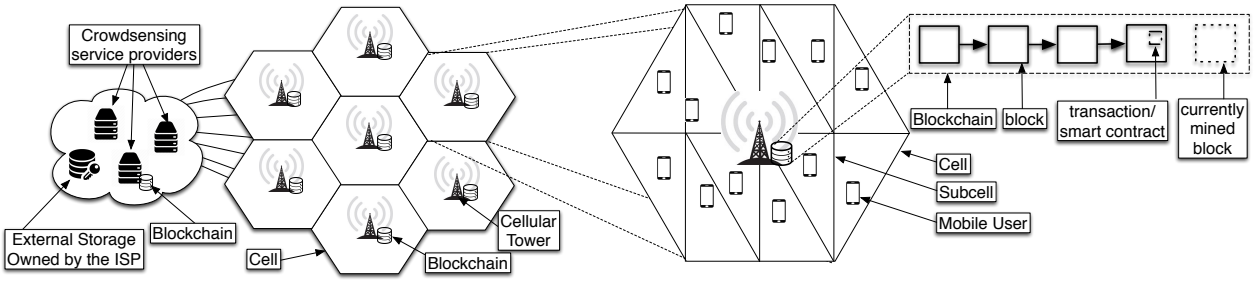


Fig. 1: The examined ecosystem. A blockchain is maintained by *(i)* the ISP and *(ii)* anyone else interested in the stored data. Smart contracts are used to coordinate the interactions between the ISP, the CSPs and the mobile users.

and for enabling small payments between mobile users in environments without internet connectivity [9].

In our scenario, we use the cellular access points of the ISP network to maintain a blockchain, but we assume that anyone (e.g., the CSPs) can participate. Transactional privacy guarantees that the identity of the creator of one transaction cannot be revealed. This functionality is used to hide users' identities. Smart contracts are software processes that are executed whenever a transaction is calling them when it is added to the blockchain. Ethereum allows any application to be deployed, using smart contracts, on the blockchain [5], [10]. For a smart contract to be executed, a certain amount of credits has to be transferred to their address. We use this feature to enforce payments. Blockchains are more preferable to servers for various reasons. First of all, they are open and append-only mechanisms that can guarantee that the stored data can not be modified. This feature guarantees the integrity of the stored data. Second, the use of the smart contracts allows anyone to examine the validity of the produced outcomes [11].

Figure 1 shows the examined architecture and the participating entities (CSPs, ISP, MUs). Cellular towers can estimate, with high accuracy, the current location of each user and for that reason, we assume that a cell can be further split into smaller areas to allow the submission of crowdsensing requests with high granularity. The ISP employs smart contracts to *(i)* give access to CSPs to the collected data they requested, *(ii)* preserve the privacy of mobile users, *(iii)* run auctions, *(iv)* pay mobile users and *(v)* get paid by the CSPs. This means that the trinity of CSPs, mobile users and the ISP interact with each other using smart contracts that are stored and executed in the blockchain. In summary, our contributions are the following:

Contributions: We address the problem of privacy preserving crowdsensing in a cost-optimal way by proposing the use of an ISP as the intermediary between CSPs and mobile users. ISP uses smart contracts over a blockchain to preserve the privacy of mobile users while ensuring the validity of their locations. As far as incentives for mobile user participation, we have designed a truthful, computationally efficient auction, called CSOPT. The cost-effectiveness of CSOPT is compared with a state-of-the-art algorithm, and the performance of the proposed smart contracts is depicted using Ethereum.

II. Related Work

Mobile users are motivated to spoof their location to preserve their privacy and potentially decrease their execution cost [12], [13], [14]. Privacy concerns might even discourage users from participating. Depending on the type of a task, the potential privacy breach changes. For example, a task that requires an MU to report the time needed to travel from one location to another by traveling at the time of the request, might lead to the disclosure of their current location and potentially sensitive addresses or even their identity through location-based attacks [15]. In the case of frequent participation, even if participants are using pseudonyms, their trajectory might reveal their sensitive locations or commutes [16] and even eventually disclose their identities [17].

Although there is high research activity on mobile crowdsensing, neither blockchain nor smart contracts have been used in the existing proposals, to the best of our knowledge. Proposed crowdsensing architectures are composed of a mobile application and a server that is responsible for the collection and processing of the sensed data. Localized analytics on the mobile devices are often performed to preserve users' privacy and reduce the amount of the data sent to the server [18]. Furthermore, similar to the deployment of smart contracts in the orchestration of the crowdsensing process, the authors of [19] develop Medusa, a framework to develop crowdsensing applications. However, the authors consider a crowdsensing application provider that is using cloud resources and do not provide any privacy guarantees to the mobile users. Similarly to this work the authors of [20] propose the ungearing of the crowdsensing provider from the physical resources that are responsible for the data gathering and processing. However they consider cloud infrastructure providers for that role, who do not provide any privacy guarantees. Liu *et. al.* [21] consider the employment of a network provider to handle the crowdsensing process but they do not consider an auction in the determination of the users' cost since they assume that the ISP will determine the credits each MU gets.

In our proposal the MUs are paid based on their costs and for which we rely on auctions. A *cost optimal auction* is an auction that minimizes the expected payments of the CSP subject to feasibility constraints [22]. In his seminal work, Myerson [22] introduces the notion of optimal auction and

designs one for selling a single unit of a single item. Our case is multiple units of multiple items (homogeneous but location specific tasks and hence we refer to it as multiple items). In economic terms, it falls under the category of *multi-unit combinatorial auctions*, which is in general hard to solve. Optimal multiple items auctions have been proposed for specific settings. For example, Cai *et. al.* [23] consider additive value settings. Iyengar and Kumar [24] design an optimal multi-unit but single item auction. Mechanism design theory has been used for crowdsensing to design incentives [25], [26], [27]. Koutsopoulos [25] designs an optimal auction for crowdsensing. However, there is no deadline or no limit on the amount of the work a participant is willing to do or any location specific tasks. Hence his work is single item multiple units. Karaliopoulos *et.al.* [28] and Yang *et. al.* [26] consider a setting the same as ours except for the fact that we offer the flexibility to the ISP to assign MUs a subset of tasks instead of a complete set of the tasks in which they show interest. This leads to cost saving to the CSP as we do not repeat any task more than required. In [28] the authors design approximate cost minimizing solutions, but do not consider the strategic behaviour of the participants. Yang *et. al.* consider designing a truthful auction for the settings very similar to ours. However, their goal is to design a computationally efficient and truthful auction. In our settings, we allow ISP to allocate an MU any subset of set of tasks in which it has shown an interest. In addition, we minimize the total expected payment made by the CSP. Another approach to offer incentives is fixed rewards rather than auction based mechanisms. For example. the incentive schemes proposed in [29], [30], [31], [32]. However, in such settings the MUs are either overpaid or there is a need for more MUs, since the payments are less than their actual cost of delivering the task. For more on game theoretic approaches on incentive design, the readers are referred to [3].

III. Mobile Crowdsensing using Blockchain

CSPs send their requests to the ISP who uses a smart contract to register the requests and collect the fees from the CSP for their requests. Then the ISP runs the auction using another smart contract to provide transparency in the selection of the proper mobile users. This smart contract forces the MUs to pay a participation fee that they will lose if they are selected and not submitted their measurements. Before the auction, the ISP creates a temporary id for each user in order to preserve the identity of the MUs. Next, the ISP uses another smart contract to collect participation proofs from the MUs and pay them. The MUs will only submit their collected data to the ISP but they will create a transaction that includes a hash of their data in order to trigger the smart contracts that pays them. Also, a fourth smart contract will give access to the CSP to the collected data. In order to execute this smart contract and get access to the collected data, the CSP has to transfer as many credits as the auction cost. The proposed smart contracts can be managed via mechanisms similar to [33]. Before going into the details of our proposal, we introduce the used notation.

A. Notation and Assumptions

We consider a set of mobile users (MUs), $\mathcal{N}$, of size $|\mathcal{N}| = n$, one crowdsensing service provider, CSP, and one Internet service provider, ISP (the model can be generalized for more than one CSPs). Whenever the CSP sends a request, CS_{req}, to the ISP with deadline D, the ISP maps the request to a set of tasks $\mathcal{T}$ and runs an auction on the appropriate cells. Each cell $\mathcal{Z}_i \in \mathcal{Z}$ is further split into areas $z_{ij} \in \mathcal{Z}_i$. Each mobile user MU_i is associated with a location, $l_i = z_{jl} \in \mathcal{Z}_j \in \mathcal{Z}$ and is able to bid for the set of tasks $\mathcal{T}_i \subset \mathcal{T} = \{T_{i1}, T_{i2}, \ldots, T_{ik_i}\}$ that it can deliver based on its current location and using the proper sensor before D. Each MU successfully completes a task with probability α. The CSP requires enough MUs at each location, in order for the probability to successfully receive the task to be at least β. Given that the mobile users need to move to the appropriate locations to do the tasks, we assume that the maximum number of tasks a user can do is k. The cost for the execution of the first task for MU_i is c_{i1}, for the second task c_{i2} and so on. We denote its cost vector by $\mathbf{c}_i \in \mathbf{C}_i$ and private information as $\theta_i = (\mathbf{c}_i, \mathcal{T}_i)$, which is called its *type* in mechanism design theory. It submits a bid $b_i = (\hat{\mathbf{c}}_i, \hat{\mathcal{T}}_i)$, where $\hat{\mathbf{c}}_i$ is its reported cost and $\hat{\mathcal{T}}_i \subset \mathcal{T}_i$ the reported tasks of interest. The ISP collects all the bids $\mathbf{b} = (b_1, b_2, \ldots, b_n) = (b_i, b_{-i})$ where b_{-i} represents the bids from all MUs except MU_i. Upon receiving $\mathbf{b}$, ISP determines the assignments, $\mathcal{A}(\mathbf{b}) = (\mathcal{AT}_1, \mathcal{AT}_2, \ldots, \mathcal{AT}_n)$, where $\mathcal{AT}_i \subset \hat{\mathcal{T}}_i$ is a set of tasks assigned to MU_i, and the payments $\mathbf{p}(\mathbf{b}) = (p_1(\mathbf{b}), p_2(\mathbf{b}), \ldots, p_n(\mathbf{b}))$. Then, the CSP is informed about the availability of the requested data.

Let $n_i = \mid \mathcal{AT}_i \mid$ denote the number of the tasks that assigned to MU_i. With these, MU_i obtains utility $u_i(\cdot)$ by participating in the crowdsensing auction. For given bids $\mathbf{b}$ and true type θ_i, u_i is given by:

$$u_i(\mathbf{b}; \theta_i) = p_i(\mathbf{b}) - \sum_{j=1}^{j=n_i} c_{ij}.$$

We drop argument $\mathbf{b}$ and just use p_i, n_i whenever it is clear from the context. We use either b_i or $(\hat{\mathbf{c}}_i, \hat{\mathcal{T}}_i)$ based on convenience in the proof and the same for θ_i and $(\mathbf{c}_i, \mathcal{T}_i)$. In our model, we assume that the MUs will not submit their bids for the tasks they cannot do. This is a valid assumption and we show how to ensure this using smart contracts. We also assume that there is enough competition between MUs and even if we exclude one MU, the request can still be served. In the next section, we explain the role of the blockchain in our model and after that, the required smart contracts in order for the model to be functional.

B. The use of Blockchain

There exist two types of interactions in our model:

1) ***Conventional****:* There are three interactions of this type. *(i)* The requests from the CSPs that contains the characteristics of the tasks (SC_{req}), *(ii)* the advertisement of the tasks from the ISP to the MUs and the initiation of the auctions *ADV*($\mathcal{T}, \mathcal{C}$), and *(iii)* the submission of the sensed data from the mobile users.

*2) **Blockchain-based**:* These interactions take the form of transactions and are stored in the blockchain. Such interactions require the interacting entities to have an account. *Transactions* are the building blocks of blockchains, represent interactions between two or more entities and are associated with some data. In its simplest form, a transaction represents the exchange of money [34], [35] but it can also be used in more complicated forms, like the one where a mobile user submits a sensor reading. There are two types of accounts, the externally owned ones (CSPs, MUs) and the smart contracts. Smart contracts are special types of accounts, which have a set of functionalities, are stored on the blockchain, and are uniquely identifiable. They also have their own storage, which can be changed whenever they are triggered by a transaction. Smart contracts allow us to have general purpose computations on the chain. Whenever such transactions are created, every miner automatically executes the contract and considers the data included in the transaction as an input. Then, the whole blockchain network operates as a distributed virtual machine. All the remaining interactions belong to this type.

C. Proposed Smart Contracts

Whenever the ISP receives a CS_{req}, it creates a transaction which is signed with the public key of the CSP. The transaction includes the timestamp of the request, the deadline D and the address of the smart contract called *Request Registration* (RR) that the CSP will call after the deadline in order to get access to the collected data on the external database of the ISP. Before the deadline, the ISP creates another smart contract, called *Data Access* (DA), and stores its address to RR. DA contains the credentials to the external database where the ISP stores the collected data. The credentials are encrypted using the public key of the CSP in order to allow only the CSP that submitted the request to get access to the collected data. When the CSP, will trigger RR, it has to create a transaction with the RR as the destination and in order for the smart contract to be executed, the CSP has to include enough credits (in the Ethereum project, these credits are called ether [5]). In this way, the CSPs have to pay a fee included by the ISP in order to get the address of DA. Then for the execution of DA, the CSP will have to pay the amount the ISP paid to the mobile users after the collection of the data. The ISP is responsible to store in RR the address of DA and the hash of the collected data for CS_{req}. If these two entries are not filled before the deadline, the RR generates a transaction from the ISP to the CSP and transfers back the credits.

The ISP, after the reception of CS_{req}, decides which are the locations of interest and broadcasts the characteristics of the tasks to the MUs on these locations (*ADV*$(\mathcal{T},\mathcal{C})$). Also, the ISP creates a temporary account in the blockchain for each of the MUs that it will be used on for this auction. Each mobile user, MU_i, submits a bid $b_i = (\hat{\mathbf{c}}_i, \hat{\mathcal{T}}_i)$ in a form of a transaction, to express its interest on executing tasks $\hat{\mathcal{T}}_i \in \mathcal{T}$, to the blockchain using its temporary address. All the bids are submitted to the designed smart contract called *Crowdsensing Optimal* (CSOPT) that produces a new transaction that contains the task assignment. The optimality of CSOPT is presented in Section IV. In this way, the ISP is not able to manipulate the bids, the CSP is also able to verify the cost of its request and the mobile users are not revealing their identity. Since each MU needs to transfer certain credits in order to trigger CSOPT, CSOPT after the production of the assignment creates a transaction and sends back to the non selected MUs the credits they spent for the auction. The selected ones will get their credits back after the completion of their tasks. If they fail to submit their tasks, they will lose their credits. Also, CSOPT triggers another contract called *Mobile User Payment* (MUP) and stores in it the produced assignment. Each mobile user that executed one or more tasks, by the end of these tasks, uploads the data to the external storage of the ISP and using a hash of them triggers the MUP smart contract that transfers the payment and the credits used for the calls of CSOPT and MUP.

Name	Type
SC_{req}	Conventional
Request Registration (RR)	Blockchain-based
Data Access (DA)	Blockchain-based
ADV$(\mathcal{T},\mathcal{C})$	Conventional
Crowdsensing Optimal (CSOPT)	Blockchain-based
Submission of Sensed Data	Conventional
Mobile User Payment (MUP)	Blockchain-based

TABLE I: List of possible interactions among the entities. For conventional interactions the ISP employs a server that receives the requests from the CSPs.

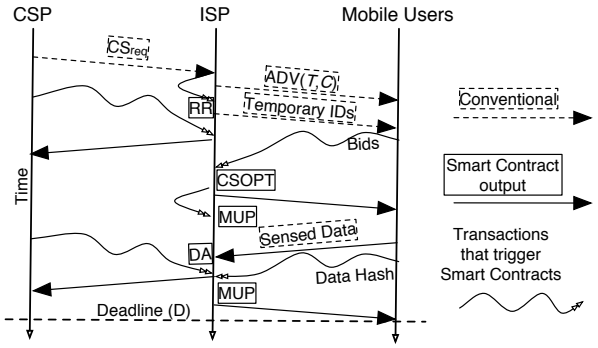


Fig. 2: Interactions between the crowdsensing service provider, the Internet service provider and the mobile users.

Table I lists and Figure 2 depicts the interactions between the participating entities. Overall, four smart contracts are used. Two between the ISP and the CSPs and two between the ISP and the MUs. These contracts guarantee that *(i)* the CSP will pay in order to get access to the collected data, *(ii)* the mobile users will get paid if they do their tasks and will loose some credits if they will not, *(iii)* the identity of the mobile users can not be revealed to the CSPs. Given that for each smart contract to be executed a transaction that has its address

as a destination needs to be mined, it is worth mentioning that we assume that the mining time of a block in the blockchain is much shorter than the deadline of the crowdsensing request.

D. Desirable Game Theoretic Properties of Auctions

We need the mobile users to report their costs as well as the tasks they can do truthfully. If the payment scheme is not designed properly, as indicated in the following example, MUs can mis-report their bids to earn more money.

Example: *Challenges in the design of a truthful auction:* Suppose there are 10 tasks and 3 interested MUs. MU_1 can do all these tasks at \$1 per task, MU_2 can do only task T_{10} at \$1.5 and MU_3 can do all these tasks at \$2.5 per task. If we decide to optimally select the set of MUs and pay them the first losing bid, all the tasks will be assigned to MU_1 who will be paid \$15 since the first losing bid is \$1.5 from MU_2 for T_{10}. However, MU_1 can misreport his bid to be \$1 per task but only for tasks T_1 to T_9. With this, he will obtain a payment of \$22.5 (2.5*9) since the first losing bid will be from MU_3 for tasks $T_1 - T_9$ and MU_2 will execute T_{10} and earn \$2.5. The total cost in this case is \$25. Thus a careful design of the auction is necessary.

If it is a best response for all the MUs to report their private information truthfully to an auction, we say the auction is *incentive compatible*. We study auctions with respect to the following two notions of incentive compatibility.

(DSIC) Dominant Strategy Incentive Compatible: An auction is called DSIC if reporting truthfully gives every MU the highest utility regardless of the bids of the other MUs.

(BIC) Bayesian Incentive Compatible: An auction is called (BIC) if reporting truthfully gives an MU highest expected utility when the other MUs are truthful, and the expectation is taken over bids of other MUs.

Apart from incentive compatibility, we also need an auction to satisfy the *individual rationality* property.

(IR) Individually Rational: An auction is called *Individually Rational* (IR) if no MU derives negative utility by participating in the auction.

Auctions could be designed with different goals. DSIC is a strong requirement that may be difficult to achieve. For that, it is a common approach in the design of auctions to enforce BIC and IR together with the desirable objective. The most popular objectives on the design of an auction are the auction to be *Allocatively Efficient* or *Cost Optimal*. An Allocatively efficient auction allocates the tasks to MUs having the least costs and achieves a socially good outcome while a cost optimal auction minimizes the cost incurred by the CSP.

In crowd-sensing, it should be ensured that each task is completed with probability β or higher. Let r be a repeat factor, that is, each task is assigned to at least r different users. The probability that the task is completed by at least one user is $1-(1-\alpha)^r \geq \beta$ or equivalently $r \geq \frac{\log(1-\beta)}{\log(1-\alpha)}$. We use X_{ij} as an indication variable with $X_{ij}=1$ if T_j is assigned to MU_i. Any auction on the examined setting needs to guarantee the following *feasibility* conditions:

$$\textstyle\sum_i X_{ij} \geq \frac{\log(1-\beta)}{\log(1-\alpha)} \quad (1)$$

$$\{T_j \mid X_{ij}=1\} \subset \mathcal{T}_i \; \forall i \quad (2)$$

With this constraints, we define allocatively efficient (AE) and cost optimal (CO) auctions as follows.

(AE) Allocatively Efficient Auction: An auction that chooses assignments that minimize the total cost incurred by MUs for every reported cost.

Optimal Auction: An auction that chooses assignments that minimize the total cost paid by the CSP.

DSIC, BIC, IR and AE are formally are formally defined in Section VIII, while the optimal auction is discussed in the next Section. In order to design a BIC and IR auction, we also need to describe the conditions on the allocation rules and payments.

Truthfulness characterization: Assuming that the cost per task is constant for all MUs. That is $\forall i \in \mathcal{N}$, $\mathbf{c}_i = (c_i, c_i, \ldots, c_i)$ and $c_i \in C_i = [\underline{c}_i, \bar{c}_i]$. Let $n_i = \sum_j X_{ij}(\mathbf{b})$ The utility of a mobile user i with bid b_i is given as,

$$\begin{aligned} u_i(b_i, b_{-i}; \theta_i) &= p_i - n_i c_i \\ U_i(b_i; \theta_i) &= P_i(b_i) - c_i N_i(b_i) \end{aligned}$$

where $N_i(b_i)$ is the expected number of tasks for MU_i where the expectation is with respect to the bids of the other agents and $P_i(b_i)$ is the expected payment.[2] We write, $P_i(b_i) = \rho_i(b_i) + \hat{c}_i N_i(b_i)$, where $\rho_i(b_i)$ is an additional incentive to report private information truthfully. Thus,

$$U_i(b_i; \theta_i) = \rho_i(b_i) - (c_i - \hat{c}_i) N_i(b_i) \quad (3)$$

Thus ρ_i represents the offered utility when all the agents are truthful. With the above offered incentive, we have the following theorem.

Theorem 1: An auction is BIC and IR if and only if $\forall i \in \mathcal{N}$,

1) $N_i(\hat{c}_i, \hat{\mathcal{T}}_i)$ is non-increasing in $\hat{c}_i \forall \hat{\mathcal{T}}_i \subset \mathcal{T}_i$.
2) $\rho_i(b_i)$ is non-negative, and non-decreasing in $\hat{k}_i$ and $\forall\ \hat{c}_i \in [\underline{c}_i, \bar{c}_i]$
3) $\rho_i(b_i) = \rho_i(\bar{c}_i, \hat{k}_i) + \int_{\hat{c}_i}^{\bar{c}_i} N_i(z, \hat{k}_i) dz$

We refer to the above statements as conditions 1, 2 and 3.

Proof: Though the key ideas in the proof are similar to [24], [22], note that our settings are quite different and we characterize the results in terms of N_is and not X_{ij}s. We present the proof in Section IX. ■

E. Ensuring the quality of Crowdsensing

It is possible for some malicious mobile users to misreport the sensed data and affect their overall quality. This makes the building of a reputation system and a careful integration of reports in the final data necessary. There have been various approaches, such as in [36], [37], proposed in the literature

[2] In general, the design of an optimal auction calls for designing the expected assignment and the expected payments for every user and every possible bid.

to limit the influence of low quality reporting. In particular, the Community Sensing Influence Limiter (CSIL) proposed by Radanovic and Faltings [36] is the most suitable for our setting. In CSIL, each MU_i has a reputation score ρ_i and data is added to the collected data with probability $\frac{\rho_i}{\rho_i+1}$. Thus, the influence of a malicious user on the aggregated data becomes limited. To build reputation scores, the ISP deploys certain trusted MUs across all cells. These MUs always perform the tasks assigned with honesty. Whenever, the ISP receives the data from trusted MUs, it updates the reputation score of each MU who has reported data for that time slot. The reputation score update function captures how much the data supplied by MU adds a value to the collected data.

F. Attack model and Defense

In order to justify that our proposal preserves users' identities and location privacy, we design an attack from a CSP that wants to find them and we explain how it fails. In order for CSPs to identify the sensitive locations (home/work) of mobile users, they need to submit requests with short deadlines at times that they expect the participants to be at such locations. However, the ISP assigns a different temporary id to each participant every time. Even if the CSP submits the same request multiple times with a short deadline in a limited geographic area and even if it is always the same participant that completes the request, the ISP will preserve her privacy since she will be assigned a different randomly selected id every time. If the id is of the same length as the addresses in Ethereum (64 bytes), the range of the possible ids is $[1, 2^{512}]$.

IV. Crowdsensing Optimal Auction

For optimal request assignments, the true costs from the MUs are needed and hence we use *mechanism design theory* to design auctions [3], [38], [39]. The goal can either be to minimize the cost incurred by the mobile users (AE auction) or to minimize the expected payment of the CSP (cost optimal auction). Note that, the CSP's goal is not to care about game theoretic property, AE, but to minimize its cost of such crowd sensing activity. Thus, we need to design a cost optimal auction for the CSP. In the examined setting, the mobile users bid for a certain set of desirable tasks and may get assigned its subset. In auction theory, this is called *combinatorial auctions*. Designing optimal combinatorial auctions for general settings is an open problem. However, there have been different attempts for specific settings [40], [41], [42]. The key difference between [40], [41] and our settings is, in their paper a mobile user either assigned the set of tasks he is interested in or nothing where as in our settings, the mobile user may get subset of its desirable tasks. In [42], the mobile user needs to submit capcity, that is how many tasks he can perform and the auction may assign any set of tasks not exceeding his capcity. In addition to combinatorial setting, we need to assign each task to multiple users to ensure high assurance on completion of tasks which is not addressed in the literature. Thus, the auction we design is categorized as an optimal multi-unit combinatorial auction. In general the characterization of an optimal combinatorial auction is an open problem. We leverage from the fact that although our setting is combinatorial, the tasks are homogeneous except from their locations. That is, a mobile user is indifferent to any constant size subset of tasks within its interested set of tasks. For example, a MU who is interested in tasks $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4$, incurs the same cost if it is assigned $\mathcal{T}_1, \mathcal{T}_2$ or $\mathcal{T}_3, \mathcal{T}_4$ or any two of these fours tasks.

We start designing an optimal auction with game theoretic properties BIC and IR. With our BIC and IR characterization result, we provide sufficient conditions for an auction to be an optimal auction in our context. Next, we study the concept of **Regularity** and prove that the optimal auction we have designed is also AE under regularity. Then we design a payment rule which along with AE allocation rule qualifies to be an optimal auction. The proposed payment rule offers difference between the cost of AE auction with their presence and absence as incentives to report their costs truthfully. That is if the cost of a MU is \$5 and the AE cost increases in his absence by \$2, it is paid \$7. We design an efficient allocation rule to determine allocation rule satisfying AE property (Algorithm 1, subroutine ALLOC-RULE). We call the proposed auction as *CSOPT*. Note that, though we set the goal to design an optimal auction with BIC and IR as constraints, CSOPT along with cost optimality also satisfies AE and DSIC.

A. CSOPT: Cost Optimal Mobile Crowdsensing Auction

An auction is called optimal, for CSP, if it minimizes the total expected payment to the MUs, is BIC and IR and is feasible [22]. That is:

$$\begin{aligned}
\text{minimize} \quad & \mathbb{E}_{\mathbf{b}} \textstyle\sum_{i\in\mathcal{N}} p_i(\mathbf{b}) \\
\text{subject to: BIC} \quad & U_i(c_i, \mathcal{T}_i; \theta_i) \geq U_i(b_i; \theta_i) \forall c_i, \forall \mathcal{T}_i \\
\text{IR} \quad & U_i(c_i, \mathcal{T}_i; \theta_i) \geq 0 \\
\text{FEASIBILITY} \quad & \textstyle\sum_i X_{ij} \geq \frac{\log(1-\beta)}{\log(1-\alpha)} \\
\text{FEASIBILITY} \quad & \{T_j \mid X_{ij} = 1\} \subset \mathcal{T}_i \; \forall i
\end{aligned}$$

Let $F_i(c_i|k_i)$ and $f_i(c_i|k_i)$ denote respectively the cumulative distribution and probability density function of cost (c_i) of MU_i given the number of tasks it can perform.

Theorem 2: Suppose the allocation rule minimizes

$$\sum_{i=1}^{n} \int_{\underline{c}_1}^{\bar{c}_1} \cdots \int_{\underline{c}_n}^{\bar{c}_n} \left(c_i + \frac{F_i(c_i|k_i)}{f_i(c_i|k_i)} \right) n_i(c_i, k_i, c_{-i}, k_{-i}) \\ f_1(c_1, k_1) \ldots f_n(c_n, k_n)\, dc_1 \ldots dc_n dk_1 dk_2 \ldots dk_n \tag{4}$$

$\forall k_i$ subject to conditions 1 and 2 of Theorem 1, Equation (1) and Equation (2). Also, suppose the payment is given by

$$P_i(c_i, k_i) = c_i N_i(c_i, k_i) + \int_{c_i}^{\bar{c}_i} N_i(z, k_i) dz \tag{5}$$

then such a payment scheme and allocation scheme constitute an optimal auction satisfying BIC and IR.

Proof: The proof is given in Section IX. ■

Algorithm 1: CSOPT

Input: $\mathcal{N}, \mathcal{T}, \hat{\mathbf{c}}, (\hat{\mathcal{T}}_i)_{i\in\mathcal{N}}, r$
Output: Allocations $\mathcal{A} = (\mathcal{AT}_1, \mathcal{AT}_2, \ldots, \mathcal{AT}_n)$ and Payments $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$.
Allocations:
$\mathcal{T} \leftarrow r\mathcal{T}$ // Make r copies of each task in $\mathcal{T}$
$\mathcal{A}$ = ALLOC-RULE($\mathcal{N}, \mathcal{T}, \hat{\mathbf{c}}, (\mathcal{T}_i)_{i\in\mathcal{N}}$)
$[p_1, p_2, \ldots, p_n]$ = PAYMENT-RULE($\mathcal{N}, \mathcal{T}, \hat{\mathbf{c}}, (\mathcal{T}_i)_{i\in\mathcal{N}}, \mathcal{A}$)

Subroutine: ALLOC-RULE($\mathring{N}, \mathring{\mathcal{T}}, \mathring{\mathbf{c}}, (\mathring{\mathcal{T}}_i)_{i\in\mathring{N}}$) ;

Input: $\langle \mathring{N}, \mathring{\mathcal{T}}, \mathring{\mathbf{c}}, (\mathring{\mathcal{T}}_i)_{i\in\mathring{N}} \rangle$
Output: Vector $\mathcal{A} = (\mathcal{AT}_1, \mathcal{AT}_2, \ldots, \mathcal{AT}_n)$ tasks assigned to each MU.
while $\mathring{\mathcal{T}} \neq \emptyset$ **do**
  Sort MUs based on cost per task for tasks in $\mathring{\mathcal{T}}$
  Add the most economic MU, say MU_i to the selected MUs
  $\mathcal{AT}_i = \mathring{\mathcal{T}} \cap \mathring{\mathcal{T}}_i$
  $\mathring{\mathcal{T}} \leftarrow \mathring{\mathcal{T}} - \mathcal{AT}_i$
$\mathcal{A} = (\mathcal{AT}_1, \mathcal{AT}_2, \ldots, \mathcal{AT}_n)$

Subroutine: PAYMENT-RULE($\mathcal{N}, \mathcal{T}, \hat{\mathbf{c}}, (\mathcal{T}_i)_{i\in\mathcal{N}}, \mathcal{A}$) ;

Input: $\langle \mathcal{N}, \mathcal{T}, \hat{\mathbf{c}}, (\mathcal{T}_i)_{i\in\mathcal{N}}, \mathcal{A} \rangle$
Output: Vector $\mathcal{P}$ of payments of each agent.
fcost = COST($\mathcal{A}$)
// COST finds out the cost of allocation $\mathcal{A}$
for $j \in \mathcal{N}$ **do**
  $\mathring{A}_j$=ALLOC-RULE($\mathcal{N} \setminus \{j\}, \mathcal{T}, \mathbf{c}_{-j}, (\hat{\mathcal{T}}_i)_{i\in\mathcal{N}\setminus\{j\}}$)
  scost=COST($\mathring{A}_j$)
  p_j= $n_j \times c_j$ + scost-fcost

(Regularity): We define the virtual cost function as

$$H_i(c_i, k_i) := c_i + \frac{F_i(c_i|k_i)}{f_i(c_i|k_i)}, \forall MU_i \in \mathcal{N}$$

We say that a type distribution is regular if $\forall i$, H_i is non-decreasing in c_i and non-increasing in k_i. Analogous to the literature on optimal auctions [24], [22], we assume regularity on our distribution type. We assume the type distributions satisfy regularity and all the MU types are independently and identically distributed (i.i.d.) over $[\underline{c_l}, \overline{c_u}] \times [\underline{k_l}, \overline{c_u}]$. We make a further assumption that the costs for all MUs are identically distributed. With these assumptions, we present the pseudocode of CSOPT in Algorithm 1.

Observation 1: Under the assumption of regularity and i.i.d. MUs, an allocatively efficient auction is an optimal solution to Equation (4) and maximizes Equation (4) for each $\mathbf{b}$.

Observation 2: Under the assumption of regularity and i.i.d. MUs, for a fixed b_{-i}, the following payment satisfies Equation (5).

$$p_i(c_i, k_i, b_{-i}) = c_i n_i(c_i, k_i, b_{-i}) + \int_{c_i}^{\overline{c}} n_i(z, k_i, b_{-i}) dz \quad (6)$$

Since we are using an AE allocation, the payment (6) can be written as:

$$p_i() = c_i n_i(c_i, k_i, b_{-i}) + V^*_{-i} - V^*$$

where V^* is the cost of AE allocation and V^*_{-i} is the cost of AE allocation if MU_i is not in the system. Observe that, keeping b_{-i} fixed, whenever MU_i increases its cost, either $n_i()$ remains the same or drops by some integer until it goes to zero. Let us assume $c_i < c_{i1} < \ldots c_{il} < \overline{c}$ are the costs at which n_i drops. Since we assume there is enough competition, eventually it should drop to zero that is $n_i(c_{il}, k_i.b_{-i}) = 0$. Precisely $c_{i1}, c_{i2}, \ldots .c_{il}$ are the costs which get added into an AE allocation when MU_i is not there in the system.

With all the above discussion, we propose our mechanism CSOPT as given in Algorithm 1. COST($\mathcal{A}$) returns the total cost of allocating tasks as described in $\mathcal{A}$. Hence $scost$ captures the total cost incurred by MU_j in optimal allocation.

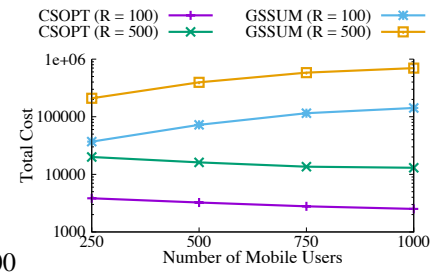
Lemma 3: CSOPT is an AE auction for the CSP.
Proof: By construction, it satisfies FEASIBILITY conditions. We need to show that it minimizes the total allocation cost. Let $\mathcal{A}_e$ be an AE allocation given bids as $\mathbf{b}$. Let $MU_1, MU_2, \ldots$ be the order in which CSOPT allocates the tasks to MUs. Let MU_i be the first MU whose allocation in CSOPT differs from that of in $\mathcal{A}_e$. Thus at least one of its tasks is assigned to MU_j $j > i$. However, $c_i \leq c_j$. Thus not awarding all $n_i()$ tasks to MU_i which are allocated by CSOPT the cost is not going to improve. Using induction, it follows that no allocation $\mathcal{A}_e$ can improve on cost of allocation over CSOPT. ■

Theorem 4: CSOPT is an optimal auction for the CSP.
Proof: It follows from Observations 1,2, Lemma 3 and Theorem 2. ■

V. Performance Evaluation

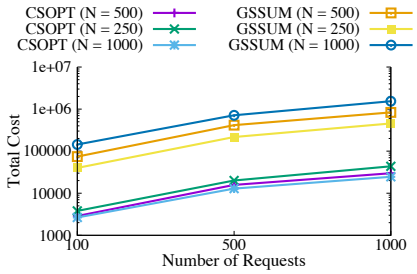
We conduct two set of experiments to depict the quality of our proposal. First, we compare the cost of the proposed auction with another state-of the-art algorithm. Next, we examine the time needs of the architecture to process crowdsensing requests. Figure 3 shows the results from the first set of experiments and Figure 4 of the second.

CSOPT: Since we have proved mathematically that CSOPT is allocatively efficient we only need to compare the cost of the allocations it produces with other state-of-the-art algorithms. For that we selected [28] because it adopts to our scenario and it is also fast in terms of time since it operates in a greedy manner. The authors named their algorithm greedy heuristic for selection under stochastic user mobility, but here we refer to it as GSSUM for short. We consider an area that is composed by a 100 by 100 grid and we randomly place mobile users on this grid. Then we generate crowdsensing requests with deadlines and we assume that each user bids for a task only if it is within a certain distance. Each user has a cost per allocated task in a range between 50 and 100. Figure 3a shows, in logarithmic scale, that the total cost (payments

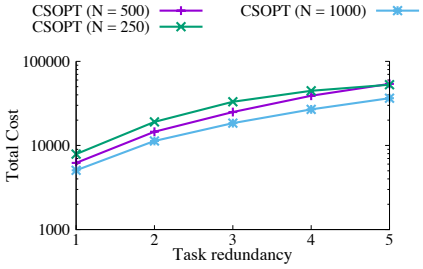


00

(a) Auction cost as function of the number of the mobile users.

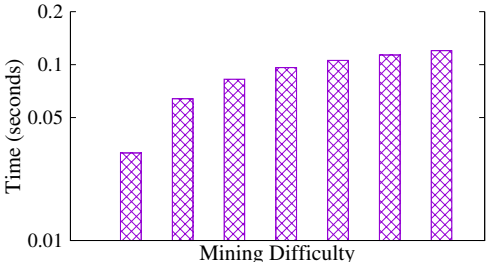


(b) Auction cost as a function of the number of the requests.

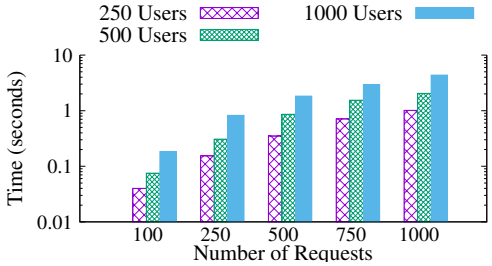


(c) Auction cost as a function of the task redundancy.

Fig. 3: Performance evaluation of CSOPT and comparison with GSSUM algorithm that was proposed by the authors of [28].



(a) The mining time of a new block for a given mining difficulty.



(b) The time needed for the execution of CSOPT for different number of users and requests.

Fig. 4: The two components that affect execution time. Mining difficulty determines the time needed for the execution of a smart contract. CSOPT cannot be executed instantaneously and for that we measure its time requirements separately.

to the MUs) of CSOPT is decreasing as the number of MUs is increasing. This result is expected because the increased competition between MUs decreases the cost per task. On the other hand, the GSSUM algorithm operates in the opposite way because whenever it selects a mobile user to assign a task to, it assigns all the tasks on which she has a bid.

Next, we compare the two algorithms in terms of the number of requests. Figure 3b shows that CSOPT is at least one order of magnitude less costly while the cost in both algorithms is increasing at the same rate. Next, in Figure 3c we show how the costs increase when the number of requests reaches 200 but there is a repeat factor $r = [1, 2, 3, 4, 5]$ that ensures that the requests will be satisfied, as explained in Section III-D. This case differs from the previous one because the requests are less disseminated throughout the whole area and the average number of the participants that can handle a request is much smaller. r in Figures 3a and 3b is 1.

Architecture: We install Ethereum in a Desktop with Intel Core i7-7700 CPU @ 3.60 GHz and 16 GB of RAM. We then measure the time required for a block to be mined for different values of mining difficulty and the time requirements of CSOPT. Figure 4 depicts these measurements. In order to produce Figure 4a we set the mining difficulty on the genesis file of Ethereum and wait for 100 blocks to be mined. Small values of mining difficulty can produce a new block every few millisecond but this will produce the generation of many empty blocks that are a waste of storage. For the time measurements of CSOPT for different numbers of MUs and crowdsensing requests, we implement the algorithm and measured its performance on the same desktop. Figure 4b shows that the time CSOPT needs to determine the task assignment and the payments increases with the number of requests and MUs. However, it does not require more than 10 seconds in the case of 1000 MUs and 1000 tasks. We denote the time requirements of CSOPT with t_{CSOPT} and the mining time of a block by t_B.

These experiments are important since all the blockchain-based interactions as described in Section III-B will have this delay. In total, any request from a CSP needs: 1 block to be mined in order to register the request (RR) while in parallel the ISP contacts the mobile users and announces the tasks (t_{ann}). If the announcement time takes more time than the mining of RR, the mining time of this block can be ignored. Next, the users bid for a predefined time period ($t_{bidding}$) and after that the CSOPT is triggered, whose termination triggers MUP. If the crowdsensing task duration t_{task} is longer than t_B, the mining of the block that is caused by the MUP is not counted in the total delay. By the end of the task execution, the users submit the collected data and DA is triggered after t_B notifies the CSP that the data have been collected. The total delay between the submission of the request and the access to the collected data is:

$$\max(t_B, t_{ann}) + t_{bidding} + t_{CSOPT} + \max(t_B, t_{task}) + t_B.$$

From this set of experiments we can conclude that if the duration of the crowdsensing tasks is in the order of tens of seconds the time overhead of using a blockchain instead of a centralised server is negligible while the benefits in terms of preserving the users privacy are high.

VI. Conclusion

In conclusion, we proposed a novel architecture for event-based spatial crowdsensing tasks that is deployed by an ISP and is based on blockchain technology. The proposed architecture employs smart contracts *(i)* that allow crowdsensing service providers to submit their requests, *(ii)* to run a cost-optimal auction for the determination of the most suitable mobile users that are interested in executing the crowdsensing tasks, *(iii)* to deal with the payments for the mobile users and *(iv)* to give access to the crowdsensing provider. The proposed architecture preserves the privacy of the mobile users in the sense that the crowdsensing provider cannot know their identity and can not derive their sensitive information such as the location of their home/work. Moreover, we have shown that the employed incentive compatible cost optimal auction that determines the selection of the mobile users that will handle each crowdsensing task, outperforms state of the art proposals when adopted to the examined setting by one order of magnitude for high numbers of mobile users and tasks.

VII. Acknowledgements

This research has been supported, in part, by projects 26211515 and 16214817 from the Research Grants Council of Hong Kong.

References

[1] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "Greengps: A participatory sensing fuel-efficient maps application," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 151–164.

[2] T. Yan, B. Hoh, D. Ganesan, K. Tracton, T. Iwuchukwu, and J.-S. Lee, "Crowdpark: A crowdsourcing-based parking reservation system for mobile phones," *Univ. of Massachusetts Amherst Tech. Report*, 2011.

[3] N. Nisan, "Introduction to mechanism design (for computer scientists)," in *Algorithmic game theory*, V. V. Vazirani, N. Nisan, T. Roughgarden, and . Tardos, Eds. Oxford University Press, 2007, ch. 9, pp. 209–242.

[4] European Commission, "Code of EU Online Rights," https://ec.europa.eu/digital-single-market/en/code-eu-online-rights, accessed April 2018.

[5] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, 2014.

[6] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 839–858.

[7] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.

[8] D. Chatzopoulos, M. Ahmadi, S. Kosta, and P. Hui, "Flopcoin: A cryptocurrency for computation offloading," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1062–1075, May 2018.

[9] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui, "Localcoin: An ad-hoc payment scheme for areas with high connectivity: Poster," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '16, 2016, pp. 365–366.

[10] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.

[11] S. Hu, C. Cai, Q. Wang, C. Wang, X. Luo, and K. Ren, "Searching an encrypted cloud meets blockchain: A decentralized, reliable and fair realization."

[12] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, Aug 2001.

[13] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful gps spoofing attacks," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 75–86.

[14] J. Yang, Y. Chen, and W. Trappe, "Detecting spoofing attacks in mobile wireless environments," in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2009, pp. 1–9.

[15] L. Pournajaf, D. A. Garcia-Ulloa, L. Xiong, and V. Sunderam, "Participant privacy in mobile crowd sensing task management: A survey of methods and challenges," *SIGMOD Rec.*, vol. 44, no. 4, pp. 23–34, 2016.

[16] J. Krumm, "Inference attacks on location tracks," *Pervasive computing*, pp. 127–143, 2007.

[17] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "De-anonymization attack on geolocated data," *Journal of Computer and System Sciences*, vol. 80, no. 8, pp. 1597–1614, 2014.

[18] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, 2011.

[19] M.-R. Ra, B. Liu, T. L. Porta, and R. Govindan, "Medusa: A Programming Framework for Crowd-Sensing Applications," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys'12)*, June 2012.

[20] G. Merlino, S. Arkoulis, S. Distefano, C. Papagianni, A. Puliafito, and S. Papavassiliou, "Mobile crowdsensing as a service: A platform for applications on top of sensing clouds," *Future Generation Computer Systems*, vol. 56, pp. 623 – 639, 2016.

[21] C. H. Liu, P. Hui, J. W. Branch, C. Bisdikian, and B. Yang, "Efficient network management for context-aware participatory sensing," in *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2011, pp. 116–124.

[22] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[23] Y. Cai, C. Daskalakis, and S. M. Weinberg, "An algorithmic characterization of multi-dimensional mechanisms," in *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, ser. STOC '12. ACM, 2012, pp. 459–478.

[24] G. Iyengar and A. Kumar, "Optimal procurement mechanisms for divisible goods with capacitated suppliers," *Review of Economic Design*, vol. 12, no. 2, pp. 129–154, 2008.

[25] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *IEEE Infocom*, 2013, pp. 1402–1410.

[26] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1732–1744, Jun. 2016.

[27] D. Zhao, X. Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 1213–1221.

[28] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *IEEE INFOCOM*, April 2015, pp. 2254–2262.

[29] G. Goel, A. Nikzad, and A. Singla, "Allocating tasks to workers with matching constraints: Truthful mechanisms for crowdsourcing markets," in *Proceedings of the 23rd International Conference on World Wide Web*, ser. WWW '14 Companion, 2014, pp. 279–280.

[30] S. Bhattacharya, G. Goel, S. Gollapudi, and K. Munagala, "Budget constrained auctions with heterogeneous items," in *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, ser. STOC '10. New York, NY, USA: ACM, 2010, pp. 379–388.

[31] G. Radanovic, B. Faltings, and R. Jurca, "Incentives for Effort in Crowdsourcing using the Peer Truth Serum," *Acm Transactions on Intelligent Systems and Technology*, vol. 7, no. 4, July 2016.

[32] G. Radanovic and B. Faltings, "Learning to Scale Payments in Crowdsourcing with PropeRBoost," in *Proceedings of the Fourth AAAI conference on Conference on Human Computation and Crowdsourcing*, 2016.

[33] Y.-C. Hu, T.-T. Lee, D. Chatzopoulos, and P. Hui, "Hierarchical interactions between ethereum smart contracts across testnets," in *Pro-*

ceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems, ser. CryBlock'18, 2018, pp. 7–12.
[34] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf
[35] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2084–2123, thirdquarter 2016.
[36] G. Radanovic and B. Faltings, "Limiting the influence of low quality information in community sensing," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, ser. AAMAS '16, 2016, pp. 873–881.
[37] P. Resnick and R. Sami, "The influence limiter: Provably manipulation-resistant recommender systems," in *Proceedings of ACM RecSys*, 2007, pp. 25–32.
[38] D. Garg, Y. Narahari, and S. Gujar, "Foundations of mechanism design: A tutorial part 1-key concepts and classical results," *Sadhana*, vol. 33, no. 2, pp. 83–130, 2008.
[39] ——, "Foundations of mechanism design: A tutorial part 2-advanced concepts and results," *Sadhana*, vol. 33, no. 2, pp. 131–174, 2008.
[40] S. Gujar and Y. Narahari, "Optimal multi-unit combinatorial auctions with single minded bidders," in *Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on*. IEEE, 2009, pp. 74–81.
[41] ——, "Optimal multi-unit combinatorial auctions," *Operational Research*, vol. 13, no. 1, pp. 27–46, 2013.
[42] S. Bhat, S. Jain, S. Gujar, and Y. Narahari, "An optimal bidimensional multi-armed bandit auction for multi-unit procurement," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 1789–1790.

VIII. Formal Definitions

Definition 1: **(Dominant Strategy Incentive Compatible):** An auction is called *Dominant Strategy Incentive Compatible* (DSIC) if reporting truthfully gives every MU the highest utility regardless of the bids of the other MUs.

Formally, $\forall i \in N, \forall \mathbf{c}_i, \hat{\mathbf{c}}_i \in \mathbf{C}_i \forall \hat{\mathcal{T}}_i \subset \mathcal{T}_i, \forall b_{-i}$,

$$u_i(\mathbf{c}_i, \mathcal{T}_i, \mathbf{b}_{-i}; \theta_i) \geq u_i(\hat{\mathbf{c}}_i, \hat{\mathcal{T}}_i, \mathbf{b}_{-i}; \theta_i).$$

Definition 2: **(Bayesian Incentive Compatible):** An auction is called *Bayesian Incentive Compatible* (BIC) if reporting truthfully gives an MU highest expected utility when the other MUs are truthful, and the expectation is taken over bids of other MUs.

Formally, $\forall i \in \mathcal{N}, \forall \hat{\mathbf{c}}_\mathbf{i}, \mathbf{c}_\mathbf{i}$,

$$U_i(\mathbf{c}_\mathbf{i}, \mathcal{T}_i; \theta_i) \geq U_i(\hat{\mathbf{c}}_\mathbf{i}, \hat{\mathcal{T}}_i; \theta_i),$$

where, $U_i(\mathbf{b}_i; \theta_i) = \mathbb{E}_{b_{-i}}[u_i(\mathbf{b}_i, \mathbf{b}_{-i}; \theta_i)]$.

Definition 3: **(Individually Rational):** An auction is called *Individually Rational* (IR) if no MU derives negative utility by participating in the auction.

Formally, $\forall i \in N, \forall \mathbf{c}_i \in \mathbf{C}_i, \mathcal{T}_i \subset \mathcal{T}$,

$$u_i(\mathbf{c_i}, \mathcal{T}_i, \mathbf{b}_{-i}; \mathbf{c_i}, \mathcal{T}_i) \geq 0$$

Definition 4: **(Allocatively Efficient (AE) Auction):** If an auction chooses assignments that minimize the total cost incurred by MUs for every reported cost, we call it an *allocatively efficient (AE)* auction.

That is, $\forall \mathbf{c}$ the auction assigns tasks such that:

$$\underset{\mathbb{X}}{\text{minimize}} \quad \textstyle\sum_{i \in \mathcal{N}} \sum_{j=1}^{j=k} c_{ij} X_{ij} \tag{7}$$

$$\text{subject to} \quad \textstyle\sum_i X_{ij} \geq \frac{\log(1-\beta)}{\log(1-\alpha)} \tag{8}$$

$$\{T_j \mid X_{ij} = 1\} \subset \mathcal{T}_i \; \forall i \tag{9}$$

and each task is assigned to at least r different mobile users.

IX. Proofs

A. Proof of Theorem 1

Proof: To prove the necessity part of the theorem, we first observe due to BIC we have,

$$U_i(\hat{c}_i, \hat{k}_i; c_i, k_i) \leq U_i(c_i, k_i; c_i, k_i) \qquad \forall (\hat{c}_i, \hat{k}_i) \text{ and } (c_i, k_i)$$
$$\implies U_i(\hat{c}_i, k_i; c_i, k_i) \leq U_i(c_i, k_i; c_i, k_i)$$

Without loss of generality, we assume $\hat{c}_i > c_i$ Rearrangement of these terms yields,

$$U_i(\hat{c}_i, k_i; c_i, k_i) = U_i(\hat{c}_i, k_i; \hat{c}_i, k_i) + (\hat{c}_i - c_i) N_i(\hat{c}_i, k_i),$$

which implies that,

$$\frac{U_i(\hat{c}_i, k_i; \hat{c}_i, k_i) - U_i(c_i, k_i; c_i, k_i)}{\hat{c}_i - c_i} \leq -N_i(\hat{c}_i, k_i).$$

Similarly using $U_i(c_i, k_i; \hat{c}_i, k_i) \leq U_i(\hat{c}_i, k_i; \hat{c}_i, k_i)$,

$$-N_i(c_i, k_i) \leq \frac{U_i(\hat{c}_i, k_i; \hat{c}_i, k_i) - U_i(c_i, k_i; c_i, k_i)}{\hat{c}_i - c_i} \leq -N_i(\hat{c}_i, k_i). \tag{10}$$

Taking limit $\hat{c}_i \to c_i$, we get,

$$\frac{\partial U_i(c_i, k_i; c_i, k_i)}{\partial c_i} = -N_i(c_i, k_i). \tag{11}$$

Equation (10) implies, $N_i(c_i, k_i)$ is non-increasing in c_i. This proves condition 1 of the theorem in the forward direction. When the worker bids truthfully, from Equation (3),

$$\rho_i(c_i, k_i) = U_i(c_i, k_i; c_i, k_i). \tag{12}$$

For BIC, Equation (11) should be true. So,

$$\rho_i(c_i, k_i) = \rho_i(\bar{c}_i, k_i) + \int_{c_i}^{\bar{c}_i} N_i(z, k_i) dz \tag{13}$$

This proves condition 3 of the theorem. BIC also requires,

$$k_i \in \text{argmax}_{\hat{k}_i} U_i(c_i, \hat{k}_i; c_i, k_i) \; \forall \; c_i \; \in \; [\underline{c}_i, \bar{c}_i]$$

This implies, $\forall c_i, \; \rho_i(c_i, k_i)$ should be non-decreasing in k_i. The IR conditions (Equation(12)) imply

$$\rho_i(c_i, k_i) \geq 0.$$

This proves condition 2 of the theorem. Thus, these three conditions are necessary for BIC and IR properties. We now prove the sufficiency. Consider

$$U_i(c_i, k_i; c_i, k_i) = \rho_i(c_i, k_i) \geq 0.$$

So the IR property is satisfied. Without loss of generality, we assume $\hat{c}_i > c_i$. The proof is similar for the case $\hat{c}_i < c_i$.

$$
\begin{aligned}
&U_i(b_i; c_i, k_i) \\
&= \rho_i(\hat{c}_i, \hat{k}_i) + (\hat{c}_i - c_i) N_i(\hat{c}_i, \hat{k}_i) && \text{(By Defn)} \\
&= \rho_i(\bar{c}_i, \hat{k}_i) + \int_{\hat{c}_i}^{\bar{c}_i} N_i(z, \hat{k}_i) dz + (\hat{c}_i - c_i) N_i(\hat{c}_i, \hat{k}_i) \\
& && \text{(By hypothesis)} \\
&= \rho_i(\bar{c}_i, \hat{k}_i) + \int_{c_i}^{\bar{c}_i} N_i(z, \hat{k}_i) dz \\
&\qquad - \int_{c_i}^{\hat{c}_i} N_i(z, \hat{k}_i) dz + (\hat{c}_i - c_i) N_i(\hat{c}_i, \hat{k}_i) \\
&\leq \rho_i(c_i, \hat{k}_i) && (N_i \text{ is non-increasing in } c_i) \\
&\leq \rho_i(c_i, k_i) && (\text{ as } \rho_i \text{ is non-decreasing in } k_i) \\
&= U_i(c_i, k_i; c_i, k_i)
\end{aligned}
$$

■

B. Proof of the Theorem 2

Proof: The auctioneer's objective is to maximize her expected utility subject to conditions BIC, IR, and Feasibility. Her objective function is:

$$
\begin{aligned}
&\sum_{i=1}^{n} \int_{\underline{c}_1}^{\bar{c}_1} \dots \int_{\underline{c}_n}^{\bar{c}_n} \int_{\underline{k}_1}^{\bar{k}_1} \dots \int_{\underline{k}_n}^{\bar{k}_n} \big[- p_i(b) \big] \\
&f_1(c_1, k_1) \dots f_n(c_n, k_n) dc_1 \dots dc_n \, dk_1 \dots dk_n \\
&= \sum_{i=1}^{n} \int_{\underline{c}_1}^{\bar{c}_1} \dots \int_{\underline{c}_n}^{\bar{c}_n} \int_{\underline{k}_1}^{\bar{k}_1} \dots \int_{\underline{k}_n}^{\bar{k}_n} \big[(-c_i + c_i) n_i(c_i, k_i, c_{-i}, k_{-i}) - p_i(b) \big] \\
&\qquad f_1(c_1, k_1) \dots f_n(c_n, k_n) dc_1 \dots dc_n \, dk_1 \dots dk_n \\
&= \sum_{i=1}^{n} \int_{\underline{c}_1}^{\bar{c}_1} \dots \int_{\underline{c}_n}^{\bar{c}_n} \int_{\underline{k}_1}^{\bar{k}_1} \dots \int_{\underline{k}_n}^{\bar{k}_n} \big(- c_i n_i(c_i, k_i, c_{-i}, k_{-i}) \big) \\
&f_1(c_1, k_1) \dots f_n(c_n, k_n) dc_1 \dots dc_n \, dk_1 \dots dk_n \\
&+ \sum_{i=1}^{n} \int_{\underline{c}_1}^{\bar{c}_1} \dots \int_{\underline{c}_n}^{\bar{c}_n} \int_{\underline{k}_1}^{\bar{k}_1} \dots \int_{\underline{k}_n}^{\bar{k}_n} \left(c_i n_i(c_i, k_i, c_{-i}, k_{-i}) - p_i(b) \right) \\
&\qquad f_1(c_1, k_1) \dots f_n(c_n, k_n) \, dc_1 \dots dc_n \, dk_1 \dots dk_n \qquad (14)
\end{aligned}
$$

The first term of Equation (14) is already same as first term in desired form of objective function of auctioneer given in Equation (4). We now use conditions (1) and (3) of Theorem 1 to arrive at the result.

$$
\begin{aligned}
&\int_{\underline{c}_1}^{\bar{c}_1} \dots \int_{\underline{c}_n}^{\bar{c}_n} \int_{\underline{k}_1}^{\bar{k}_1} \dots \int_{\underline{k}_n}^{\bar{k}_n} \big(c_i n_i(.) - p_i(b) \big) \\
&f_1(c_1, k_1) \dots f_n(c_n, k_n) dc_1 \dots dc_n \, dk_1 \dots dk_n \\
&= - \int_{\underline{k}_i}^{\bar{k}_i} \int_{\underline{c}_i}^{\bar{c}_i} \rho_i(c_i, k_i) f_i(c_i, q_i) dc_i \, dk_i \\
& && \text{(Integrating out } b_{-i}) \\
&= - \int_{\underline{k}_i}^{\bar{k}_i} \int_{\underline{c}_i}^{\bar{c}_i} \left(\rho_i(\bar{c}_i, k_i) + \int_{c_i}^{\bar{c}_i} N_i(z, k_i) dz \right) f_i(c_i, k_i) dc_i \, dk_i \\
& && \text{(As we need truthfulness)} \\
&= - \int_{\underline{k}_i}^{\bar{k}_i} \int_{\underline{c}_i}^{\bar{c}_i} \rho_i(\bar{c}_i, k_i) f_i(c_i, k_i) dc_i \, dk_i \\
&\qquad - \int_{\underline{k}_i}^{\bar{k}_i} \int_{\underline{c}_i}^{\bar{c}_i} N_i(z, k_i) dz \int_{\underline{c}_i}^{z} f_i(c_i | k_i) dc_i \; f_i(k_i) dk_i \\
& && \text{(Changing order of integration)} \\
&= - \int_{\underline{k}_i}^{\bar{k}_i} \int_{\underline{c}_i}^{\bar{c}_i} \rho_i(\bar{c}_i, k_i) f_i(c_i, k_i) dc_i \, dk_i \\
&\qquad - \int_{\underline{k}_i}^{\bar{k}_i} \int_{\underline{c}_i}^{\bar{c}_i} N_i(z, k_i) F_i(z | k_i) dz f_i(k_i) dk_i \\
&= - \int_{\underline{k}_i}^{\bar{k}_i} \int_{\underline{c}_i}^{\bar{c}_i} \rho_i(\bar{c}_i, k_i) f_i(c_i, k_i) dc_i \, dk_i \\
&\qquad - \int_{\underline{k}_i}^{\bar{k}_i} \int_{\underline{c}_i}^{\bar{c}_i} N_i(z, k_i) \frac{F_i(c_i | k_i)}{f_i(c_i | k_i)} f_i(c_i, k_i) dz \, dk_i \qquad (15)
\end{aligned}
$$

The last step is obtained by relabeling the variable of integration and simplifying.

Here, $\rho_i(\bar{c}_i, k_i)$ denotes the utility of a MU_i when its true type is $(\bar{c}_i, k_i)$. With this type profile, the auctioneer by paying $\bar{c}_i$ can ensure both IR and IC, hence we can set $\rho_i(\bar{c}_i, k_i) = 0, \forall k_i \in [\underline{k}_i, \bar{k}_i]$. Applying this in the above equation and simplifying we get that the objective function of auctioneer is same in form to Equation (4). Consider Equation (15) and set $\rho_i(\bar{c}_i, k_i) = 0$ and simplification yields Equation (5). By construction, the mechanism is BIC and IR. By hypothesis, as the auctioneer's objective is maximized, the mechanism is optimal. ■